

Technical University of Denmark



## The Conic Benchmark Format

### Version 1 - Technical Reference Manual

**Friberg, Henrik A.**

*Publication date:*  
2014

*Document Version*  
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

*Citation (APA):*  
Friberg, H. A. (2014). The Conic Benchmark Format: Version 1 - Technical Reference Manual. DTU Wind Energy. (DTU Wind Energy E; No. 0047).

## DTU Library

### Technical Information Center of Denmark

---

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# The Conic Benchmark Format

Version 1 – Technical Reference Manual



## Department of Wind Energy E Report 2014

Henrik A. Friberg  
*PhD Candidate at the Technical University of Denmark,  
Department of Wind Energy, and employee at MOSEK ApS.  
E-mail: haf@mosek.com.*

DTU Wind Energy – Report E-0047

ISBN 978-87-92896-84-1

February 2014

**DTU Wind Energy**  
Department of Wind Energy

---



### **Abstract**

This document constitutes the technical reference manual of the Conic Benchmark Format with file extension: `.cbf` or `.CBF`. It unifies linear, second-order cone (also known as conic quadratic) and semidefinite optimization with mixed-integer variables. The format has been designed with benchmark libraries in mind, and therefore focuses on compact and easily parsable representations. The problem structure is separated from the problem data, and the format moreover facilitate benchmarking of hotstart capability through sequences of changes.

### **Acknowledgements**

Henrik A. Friberg is the creator of the CBF format and the author of this document. His work would never have been accomplished to the same degree of quality without the in-depth review and helpful comments from Erling D. Andersen and Mathias Stolpe. The ideas for a conic input format presented by Imre Pólik has been a great source of inspiration.

### **Funding**

Henrik A. Friberg was funded by MOSEK ApS, and the Danish Ministry of Science, Innovation and Higher Education through the Industrial PhD project "Combinatorial Optimization over Second-Order Cones and Industrial Applications".

# Contents

<b>1</b>	<b>Minimal working example</b>	<b>4</b>
<b>2</b>	<b>The structure of CBF files</b>	<b>5</b>
2.1	Information items . . . . .	5
2.2	Information groups and their ordering . . . . .	5
2.3	Embedded hotstart-sequences . . . . .	5
2.4	File encoding and line width restrictions . . . . .	6
2.5	Comment-line and whitespace rules . . . . .	6
<b>3</b>	<b>How instances are specified</b>	<b>7</b>
3.1	Problem structure . . . . .	7
3.2	Problem data . . . . .	9
<b>A</b>	<b>List of cones</b>	<b>10</b>
<b>B</b>	<b>List of keywords</b>	<b>11</b>
<b>C</b>	<b>Examples</b>	<b>16</b>
C.1	Mixing linear, second-order and semidefinite cones . . . . .	16
C.2	Mixing semidefinite variables and linear matrix inequalities . . .	18
C.3	Optimization over a sequence of objectives . . . . .	20

## 1 Minimal working example

The conic optimization problem (1), has three variables in a quadratic cone  $Q^3$  - first one is integer - and an affine expression in domain  $\{0\}$  (equality constraint).

$$\begin{aligned} & \text{minimize} && 5.1 x_0 \\ & \text{subject to} && 6.2 x_1 + 7.3 x_2 - 8.4 \in \{0\} \\ & && x \in Q^3, x_0 \in \mathbb{Z}. \end{aligned} \tag{1}$$

Its formulation in the Conic Benchmark Format begins with the version of the CBF format used, to safeguard against later revisions.

VER

1

Next follows the problem structure, consisting of the objective sense, the number and domain of variables, the indices of integer variables, and the number and domain of scalar-valued affine expressions (i.e., the equality constraint).

OBJSENSE

MIN

VAR

3 1

Q 3

INT

1

0

CON

1 1

L= 1

Finally follows the problem data, consisting of the coefficients of the objective, the coefficients of the constraints, and the constant terms of the constraints. All data is specified on a sparse coordinate form.

OBJCOORD

1

0 5.1

ACCOORD

2

0 1 6.2

0 2 7.3

BCCOORD

1

0 -8.4

This concludes the example! Please see Section 2 and Section 3 for details about the document structure and use of keywords.

## 2 The structure of CBF files

This section defines how information is written in the CBF format, without being specific about the type of information being communicated.

### 2.1 Information items

The format is composed as a list of information items. The first line of an information item is the **KEYWORD**, revealing the type of information provided. The second line - of some keywords only - is the **HEADER**, typically revealing the size of information that follows. The remaining lines are the **BODY** holding the actual information to be specified.

```
KEYWORD
BODY
```

```
KEYWORD
HEADER
BODY
```

The **KEYWORD** specifies how of each line in the **HEADER** and **BODY** is structured. Moreover, the number of lines in the **BODY** is decidable either from the **KEYWORD**, the **HEADER**, or from another information item required to precede it.

### 2.2 Information groups and their ordering

All information items belong to exactly one of the three groups of information. These information groups, and the order they must appear in, is:

1. File format.
2. Problem structure.
3. Problem data.

The first group, file format, provides information on how to interpret the file. It is currently limited to the keyword **VER**, specifying the version of the CBF format in use. The second group, problem structure, provides the information needed to deduce the type and size of the problem instance. Finally, the third group, problem data, specifies the coefficients and constants of the problem instance.

### 2.3 Embedded hotstart-sequences

A sequence of problem instances, based on the same problem structure, is allowed within a single file. This is facilitated via the **CHANGE** keyword used within the problem data information group, as a separator between the information items of each instance. The information items following a **CHANGE** keyword is appending to, or changing (e.g., setting coefficients back to their default value of zero), the problem data of the preceding instance.

The sequence is intended for benchmarking of hotstart capability, where the solvers can reuse their internal state and solution (subject to the achieved accuracy) as warmpoint for the succeeding instance. Whenever this feature is unsupported or undesired, the keyword **CHANGE** should be interpreted as the end of file.

## 2.4 File encoding and line width restrictions

The format is based on the US-ASCII printable character set with two extensions as listed below. Note, by definition, that none of these extensions can be misinterpreted as printable US-ASCII characters:

- A line feed marks the end of a line, carriage returns are ignored.
- Comment-lines may contain unicode characters in UTF-8 encoding.

The line width is restricted to 512 bytes, with 3 bytes reserved for the potential carriage return, line feed and null-terminator.

Integers and floating point numbers must follow the ISO C decimal string representation in the standard “C” locale. The format does not impose restrictions on the magnitude of, or number of significant digits in, numeric data, but the use of 64-bit integers and 64-bit IEEE 754 floating point numbers should be sufficient to avoid loss of precision.

## 2.5 Comment-line and whitespace rules

The format allows single-line comments respecting the following rule:

- Lines having first byte equal to '#' (US-ASCII 35) are comments, and should be ignored. Comments are only allowed between information items.

Given that a line is not a comment-line, whitespace characters should be handled according to the following rules:

- Leading and trailing whitespace characters should be ignored.
- The separator between multiple pieces of information on one line, is either one or more whitespace characters.
- Lines containing only whitespace characters are empty, and should be ignored. Empty lines are only allowed between information items.

### 3 How instances are specified

This section defines the spectrum of conic optimization problems that can be formulated in terms of the keywords of the CBF format.

#### 3.1 Problem structure

Conic optimization problems consist of variables, constraints and one objective function. In the CBF format, these are defined as follows.

- Variables are either scalar-valued (part of a vector restricted to a cone), or matrix-valued (restricted to be symmetric positive semidefinite). These are referred to as the scalar variables,  $x_j$  for  $j \in \mathcal{J}$ , and PSD variables,  $X_j$  for  $j \in \mathcal{J}^{PSD}$ . Only scalar variables can be integer.
- Constraints are affine expressions of the variables, either scalar-valued (part of a vector restricted to a cone), or matrix-valued (restricted to be symmetric positive semidefinite). These are thus referred to as the scalar constraints, with affine expressions  $g_i$  for  $i \in \mathcal{I}$ , and PSD constraints, with affine expressions  $G_i$  for  $i \in \mathcal{I}^{PSD}$ .
- The objective is a scalar-valued affine expression of the variables, either to be minimized or maximized. We refer to this expression as  $g^{obj}$ .

The problem structure defines the objective sense, whether it is minimization and maximization, using the keyword `OBJSENSE` (follow the hyperlink or see Appendix B). It also defines the index sets,  $\mathcal{J}$ ,  $\mathcal{J}^{PSD}$ ,  $\mathcal{I}$  and  $\mathcal{I}^{PSD}$ , which are all numbered from zero,  $\{0, 1, \dots\}$ , and empty until explicitly constructed.

- Scalar variables are constructed in vectors restricted to a conic domain, such as  $(x_0, x_1) \in \mathbb{R}_+^2$ ,  $(x_2, x_3, x_4) \in Q^3$ , etc. In terms of the Cartesian product, this generalizes to  $x \in K_1^{n_1} \times K_2^{n_2} \times \dots \times K_k^{n_k}$ , which in the CBF format becomes

```
VAR
n k
K1 n1
K2 n2
...
Kk nk
```

where  $n$  is the total number of scalar variables. The list of supported cones is found in Appendix A. Integrality of scalar variables can be specified afterwards, as in the minimal working example of Section 1, using the keyword `INT` (follow the hyperlink or see Appendix B).

- PSD variables are constructed one-by-one. That is,  $X_j \succeq \mathbf{0}^{n_j \times n_j}$  for  $j \in \mathcal{J}^{PSD}$ , construct matrix-valued variables of size  $n_j \times n_j$  restricted



to be symmetric positive semidefinite. In the CBF format, this list of constructions becomes

```
PSDVAR
N
n1
n2
...
nN
```

where  $N$  is the total number of PSD variables.

- Scalar constraints are constructed in vectors restricted to a conic domain, such as  $(g_0, g_1) \in \mathbb{R}_+^2$ ,  $(g_2, g_3, g_4) \in Q^3$ , etc. In terms of the Cartesian product, this generalizes to  $g \in K_1^{m_1} \times K_2^{m_2} \times \dots \times K_k^{m_k}$ , which in the CBF format becomes

```
CON
m k
K1 m1
K2 m2
...
Kk mk
```

where  $m$  is the total number of scalar constraints. The list of supported cones is found in Appendix A.

- PSD constraints are constructed one-by-one. That is,  $G_i \succeq \mathbf{0}^{m_i \times m_i}$  for  $i \in \mathcal{I}^{PSD}$ , construct matrix-valued affine expressions of size  $m_i \times m_i$  restricted to be symmetric positive semidefinite. In the CBF format, this list of constructions becomes

```
PSDCON
M
m1
m2
...
mM
```

where  $M$  is the total number of PSD constraints.

With the objective sense, variables (with integer indications) and constraints, the definitions of the many affine expressions follow in problem data.

*Keywords covered in this section:*

```
OBJSENSE - Define the objective sense.
VAR       - Construct the scalar variables.
INT       - Put integer requirements on a selected subset of scalar variables.
PSDVAR    - Construct the PSD variables.
CON       - Construct the scalar constraints.
PSDCON    - Construct the PSD constraints.
```

### 3.2 Problem data

The problem data defines the coefficients and constants of the affine expressions of the problem instance. These are considered zero until explicitly defined, implying that instances with no keywords from this information group are, in fact, valid. Duplicated or conflicting information is a failure to comply with the standard. Consequently, two coefficients written to the same position in a matrix (or to transposed positions in a symmetric matrix) is an error.

The affine expressions of the objective,  $g^{obj}$ , of the scalar constraints,  $g_i$ , and of the PSD constraints,  $G_i$ , are defined separately. The following notation uses the standard trace inner product for matrices,  $\langle X, Y \rangle = \sum_{i,j} X_{ij}Y_{ij}$ .

- The affine expression of the objective is defined as

$$g^{obj} = \sum_{j \in \mathcal{J}^{PSD}} \langle F_j^{obj}, X_j \rangle + \sum_{j \in \mathcal{J}} a_j^{obj} x_j + b^{obj},$$

in terms of the symmetric matrices,  $F_j^{obj}$ , and scalars,  $a_j^{obj}$  and  $b^{obj}$ . These are specified on a sparse coordinate form using keywords OBJFCOORD, OBJACOORD, OBJBCOORD (follow the hyperlinks or see Appendix B).

- The affine expressions of the scalar constraints are defined, for  $i \in \mathcal{I}$ , as

$$g_i = \sum_{j \in \mathcal{J}^{PSD}} \langle F_{ij}, X_j \rangle + \sum_{j \in \mathcal{J}} a_{ij} x_j + b_i,$$

in terms of the symmetric matrices,  $F_{ij}$ , and scalars,  $a_{ij}$  and  $b_i$ . These are specified on a sparse coordinate form using keywords FCOORD, ACOORD, and BCOORD (follow the hyperlinks or see Appendix B).

- The affine expressions of the PSD constraints are defined, for  $i \in \mathcal{I}^{PSD}$ , as

$$G_i = \sum_{j \in \mathcal{J}} x_j H_{ij} + D_i,$$

in terms of the symmetric matrices,  $H_{ij}$  and  $D_i$ . These are specified on a sparse coordinate form using keywords HCOORD and DCOORD (follow the hyperlinks or see Appendix B).

*Keywords covered in this section:*

OBJFCOORD - Define the affine expression of the objective.

OBJACOORD

OBJBCOORD

FCOORD - Define the affine expressions of the scalar constraints.

ACCOORD

BCCOORD

HCOORD - Define the affine expressions of the PSD constraints.

DCCOORD

## A List of cones

The format uses an explicit syntax for symmetric positive semidefinite cones as shown in Section 3.1. For scalar variables and constraints, constructed in vectors, the supported conic domains and their minimum sizes are given as follows.

- **Free domain**

CBF name F: A cone in the linear family defined by

$$\{x \in \mathbb{R}^n\}, \text{ for } n \geq 1.$$

- **Positive orthant**

CBF name L+: A cone in the linear family defined by

$$\{x \in \mathbb{R}^n \mid x_j \geq 0 \text{ for } j = 1, \dots, n\}, \text{ for } n \geq 1.$$

- **Negative orthant**

CBF name L-: A cone in the linear family defined by

$$\{x \in \mathbb{R}^n \mid x_j \leq 0 \text{ for } j = 1, \dots, n\}, \text{ for } n \geq 1.$$

- **Fixpoint zero**

CBF name L=: A cone in the linear family defined by

$$\{x \in \mathbb{R}^n \mid x_j = 0 \text{ for } j = 1, \dots, n\}, \text{ for } n \geq 1.$$

- **Quadratic cone**

CBF name Q: A cone in the second-order cone family defined by

$$\left\{ \begin{pmatrix} p \\ x \end{pmatrix} \in \mathbb{R} \times \mathbb{R}^{n-1}, p^2 \geq x^T x, p \geq 0 \right\}, \text{ for } n \geq 2.$$

- **Rotated quadratic cone**

CBF name QR: A cone in the second-order cone family defined by

$$\left\{ \begin{pmatrix} p \\ q \\ x \end{pmatrix} \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2}, 2pq \geq x^T x, p \geq 0, q \geq 0 \right\}, \text{ for } n \geq 2.$$

## B List of keywords

All keywords are case sensitive and may not appear more than once in any instance specification. In summary, by information group, they are given as:

- **File format**  
VER
- **Problem structure**  
OBJSENSE, PSDVAR, VAR, INT, PSDCON, CON
- **Problem data**  
OBJFCOORD, OBJACCOORD, OBJBCOORD  
FCOORD, ACOORD, BCOORD  
HCOORD, DCOORD  
CHANGE

The information groups must be ordered as specified in Section 2.2. All keywords, and their ordering within their information group, are optional unless explicitly stated as a remark to the respective keyword.

### VER

*The version of the Conic Benchmark Format used to write the file.*

HEADER    None.

BODY        One line formatted as:  
             INT  
             This is the version number.

#### Remarks:

Must appear exactly once in a file, as the first keyword.

### OBJSENSE

*Define the objective sense.*

HEADER    None.

BODY        One line formatted as:  
             STR  
             Having MIN indicates minimize, and MAX indicates maximize.  
             Capital letters are required.

#### Remarks:

Must appear exactly once in a file.

## PSDVAR

*Construct the PSD variables.*

HEADER    One line formatted as:  
          INT  
          This is the number of PSD variables in the problem.

BODY      A list of lines formatted as  
          INT  
          This indicates the number of rows (equal to the number of columns) in the matrix-valued PSD variable. The number of lines should match the number stated in the header.

## VAR

*Construct the scalar variables.*

HEADER    One line formatted as:  
          INT INT  
          This is the number of scalar variables, followed by the number of conic domains they are restricted to.

BODY      A list of lines formatted as  
          STR INT  
          This indicates the cone name (see Appendix A), and the number of scalar variables restricted to this cone. These numbers should accumulate to the number of scalar variables stated first in the header. The number of lines should match the second number stated in the header.

## INT

*Put integer requirements on a selected subset of scalar variables.*

HEADER    One line formatted as:  
          INT  
          This is the number of integer scalar variables in the problem.

BODY      A list of lines formatted as  
          INT  
          This indicates the scalar variable index  $j \in \mathcal{J}$ . The number of lines should match the number stated in the header.

### Remarks:

Can only be used after these keywords: VAR

## PSDCON

*Construct the PSD constraints.*

- HEADER    One line formatted as:  
          INT  
          This is the number of PSD constraints in the problem.
- BODY      A list of lines formatted as  
          INT  
          This indicates the number of rows (equal to the number of columns) in the matrix-valued affine expression of the PSD constraint. The number of lines should match the number stated in the header.

### Remarks:

Can only be used after these keywords: PSDVAR, VAR

## CON

*Construct the scalar constraints.*

- HEADER    One line formatted as:  
          INT INT  
          This is the number of scalar constraints, followed by the number of conic domains they restrict to.
- BODY      A list of lines formatted as  
          STR INT  
          This indicates the cone name (see Appendix A), and the number of affine expressions restricted to this cone. These numbers should accumulate to the number of scalar constraints stated first in the header. The number of lines should match the second number stated in the header.

### Remarks:

Can only be used after these keywords: PSDVAR, VAR

## OBJFCOORD

*Input sparse coordinates (quadruplets) to define the symmetric matrices,  $F_j^{obj}$ , as used in the objective.*

- HEADER    One line formatted as:  
          INT  
          This is the number of coordinates to be specified.
- BODY      A list of lines formatted as  
          INT INT INT REAL  
          This indicates the PSD variable index  $j \in \mathcal{J}^{PSD}$ , the row index, the column index and the coefficient value. The number of lines should match the number stated in the header.

## OBJACOORD

*Input sparse coordinates (pairs) to define the scalars,  $a_j^{obj}$ , as used in the objective.*

HEADER    One line formatted as:  
          INT  
          This is the number of coordinates to be specified.

BODY        A list of lines formatted as  
          INT REAL  
          This indicates the scalar variable index  $j \in \mathcal{J}$  and the coefficient value. The number of lines should match the number stated in the header.

## OBJBCOORD

*Input the scalar,  $b^{obj}$ , as used in the objective.*

HEADER    None.

BODY        One line formatted as:  
          REAL  
          This indicates the coefficient value.

## FCOORD

*Input sparse coordinates (quintuplets) to define the symmetric matrices,  $F_{ij}$ , as used in the scalar constraints.*

HEADER    One line formatted as:  
          INT  
          This is the number of coordinates to be specified.

BODY        A list of lines formatted as  
          INT INT INT INT REAL  
          This indicates the scalar constraint index  $i \in \mathcal{I}$ , the PSD variable index  $j \in \mathcal{J}^{PSD}$ , the row index, the column index and the coefficient value. The number of lines should match the number stated in the header.

## ACOORD

*Input sparse coordinates (triplets) to define the scalars,  $a_{ij}$ , as used in the scalar constraints.*

HEADER    One line formatted as:  
          INT  
          This is the number of coordinates to be specified.

BODY        A list of lines formatted as  
          INT INT REAL  
          This indicates the scalar constraint index  $i \in \mathcal{I}$ , the scalar variable index  $j \in \mathcal{J}$  and the coefficient value. The number of lines should match the number stated in the header.

## BCOORD

*Input sparse coordinates (pairs) to define the scalars,  $b_i$ , as used in the scalar constraints.*

HEADER    One line formatted as:  
          INT  
          This is the number of coordinates to be specified.

BODY        A list of lines formatted as  
              INT REAL  
              This indicates the scalar constraint index  $i \in \mathcal{I}$  and the coefficient value. The number of lines should match the number stated in the header.

## HCOORD

*Input sparse coordinates (quintuplets) to define the symmetric matrices,  $H_{ij}$ , as used in the PSD constraints.*

HEADER    One line formatted as:  
          INT  
          This is the number of coordinates to be specified.

BODY        A list of lines formatted as  
              INT INT INT INT REAL  
              This indicates the PSD constraint index  $i \in \mathcal{I}^{PSD}$ , the scalar variable index  $j \in \mathcal{J}$ , the row index, the column index and the coefficient value. The number of lines should match the number stated in the header.

## DCOORD

*Input sparse coordinates (quadruplets) to define the symmetric matrices,  $D_i$ , as used in the PSD constraints.*

HEADER    One line formatted as:  
          INT  
          This is the number of coordinates to be specified.

BODY        A list of lines formatted as  
              INT INT INT REAL  
              This indicates the PSD constraint index  $i \in \mathcal{I}^{PSD}$ , the row index, the column index and the coefficient value. The number of lines should match the number stated in the header.

## CHANGE

*Start of a new instance specification based on changes to the previous.*

HEADER    None.

BODY        None.

### Remarks:

Can be interpreted as the end of file when the hotstart-sequence is unsupported or undesired.



## C Examples

### C.1 Mixing linear, second-order and semidefinite cones

The conic optimization problem (2), has a semidefinite cone, a quadratic cone over unordered subindices, and two equality constraints.

$$\begin{aligned}
 & \text{minimize} && \left\langle \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}, X_1 \right\rangle + x_1 \\
 & \text{subject to} && \left\langle \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, X_1 \right\rangle + x_1 &= 1.0, \\
 & && \left\langle \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, X_1 \right\rangle + x_0 + x_2 &= 0.5, \\
 & && x_1 \geq \sqrt{x_0^2 + x_2^2}, \\
 & && X_1 \succeq \mathbf{0}.
 \end{aligned} \tag{2}$$

The equality constraints are easily rewritten to the conic form,  $(g_0, g_1) \in \{0\}^2$ , by moving constants such that the right-hand-side becomes zero. The quadratic cone does not fit under the **VAR** keyword in this variable permutation, however, as opposed to the minimal working example (1). Instead, it takes a scalar constraint  $(g_2, g_3, g_4) = (x_1, x_0, x_2) \in Q^3$ , with scalar variables constructed as  $(x_0, x_1, x_2) \in \mathbb{R}^3$ . Its formulation in the CBF format is written in verbatim.

```

# File written using this version of the Conic Benchmark Format:
#       | Version 1.
VER
1

# The sense of the objective is:
#       | Minimize.
OBJSENSE
MIN

# One PSD variable of this size:
#       | Three times three.
PSDVAR
1
3

# Three scalar variables in this one conic domain:
#       | Three are free.
VAR
3 1
F 3

# Five scalar constraints with affine expressions in two conic domains:
#       | Two are fixed to zero.

```

```

#      | Three are in conic quadratic domain.
CON
5 2
L= 2
Q 3

# Five coordinates in  $F^{\{obj\}}_j$  coefficients:
#      |  $F^{\{obj\}}[0][0,0] = 2.0$ 
#      |  $F^{\{obj\}}[0][1,0] = 1.0$ 
#      | and more...
OBJFCOORD
5
0 0 0 2.0
0 1 0 1.0
0 1 1 2.0
0 2 1 1.0
0 2 2 2.0

# One coordinate in  $a^{\{obj\}}_j$  coefficients:
#      |  $a^{\{obj\}}[1] = 1.0$ 
OBJACOORD
1
1 1.0

# Nine coordinates in  $F_{ij}$  coefficients:
#      |  $F[0,0][0,0] = 1.0$ 
#      |  $F[0,0][1,1] = 1.0$ 
#      | and more...
FCOORD
9
0 0 0 0 1.0
0 0 1 1 1.0
0 0 2 2 1.0
1 0 0 0 1.0
1 0 1 0 1.0
1 0 2 0 1.0
1 0 1 1 1.0
1 0 2 1 1.0
1 0 2 2 1.0

# Six coordinates in  $a_{ij}$  coefficients:
#      |  $a[0,1] = 1.0$ 
#      |  $a[1,0] = 1.0$ 
#      | and more...
ACOORD
6
0 1 1.0
1 0 1.0
1 2 1.0
2 1 1.0
3 0 1.0
4 2 1.0

# Two coordinates in  $b_i$  coefficients:
#      |  $b[0] = -1.0$ 
#      |  $b[1] = -0.5$ 
BCOORD

```

```

2
0 -1.0
1 -0.5

```

## C.2 Mixing semidefinite variables and linear matrix inequalities

The standard forms in semidefinite optimization are usually based either on semidefinite variables or linear matrix inequalities. In the CBF format, both forms are supported and can even be mixed as shown in (3).

$$\begin{aligned}
& \text{minimize} && \left\langle \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X_1 \right\rangle + x_1 + x_2 + 1 \\
& \text{subject to} && \left\langle \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, X_1 \right\rangle - x_1 - x_2 \geq 0.0, \\
& && x_1 \begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix} + x_2 \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \succeq \mathbf{0}, \\
& && X_1 \succeq \mathbf{0}.
\end{aligned} \tag{3}$$

Its formulation in the CBF format is written in verbatim.

```

# File written using this version of the Conic Benchmark Format:
#   | Version 1.
VER
1

# The sense of the objective is:
#   | Minimize.
OBJSENSE
MIN

# One PSD variable of this size:
#   | Two times two.
PSDVAR
1
2

# Two scalar variables in this one conic domain:
#   | Two are free.
VAR
2 1
F 2

# One PSD constraint of this size:
#   | Two times two.
PSDCON
1
2

```

```

# One scalar constraint with an affine expression in this one conic domain:
#   | One is greater than or equal to zero.
CON
1 1
L+ 1

# Two coordinates in  $F^{\{obj\}}_j$  coefficients:
#   |  $F^{\{obj\}}[0][0,0] = 1.0$ 
#   |  $F^{\{obj\}}[0][1,1] = 1.0$ 
OBJFCOORD
2
0 0 0 1.0
0 1 1 1.0

# Two coordinates in  $a^{\{obj\}}_j$  coefficients:
#   |  $a^{\{obj\}}[0] = 1.0$ 
#   |  $a^{\{obj\}}[1] = 1.0$ 
OBJACOORD
2
0 1.0
1 1.0

# One coordinate in  $b^{\{obj\}}$  coefficient:
#   |  $b^{\{obj\}} = 1.0$ 
OBJBCOORD
1.0

# One coordinate in  $F_{ij}$  coefficients:
#   |  $F[0,0][1,0] = 1.0$ 
FCOORD
1
0 0 1 0 1.0

# Two coordinates in  $a_{ij}$  coefficients:
#   |  $a[0,0] = -1.0$ 
#   |  $a[0,1] = -1.0$ 
ACCOORD
2
0 0 -1.0
0 1 -1.0

# Four coordinates in  $H_{ij}$  coefficients:
#   |  $H[0,0][1,0] = 1.0$ 
#   |  $H[0,0][1,1] = 3.0$ 
#   | and more...
HCOORD
4
0 0 1 0 1.0
0 0 1 1 3.0
0 1 0 0 3.0
0 1 1 0 1.0

# Two coordinates in  $D_i$  coefficients:
#   |  $D[0][0,0] = -1.0$ 
#   |  $D[0][1,1] = -1.0$ 
DCOORD

```

```

2
0 0 0 -1.0
0 1 1 -1.0

```

### C.3 Optimization over a sequence of objectives

The linear optimization problem (4), is defined for a sequence of objectives such that hotstarting from one to the next might be advantages.

$$\begin{aligned}
& \text{maximize}_k && g_k^{obj} \\
& \text{subject to} && 50x_0 + 31 \leq 250, \\
& && 3x_0 - 2x_1 \geq -4, \\
& && x \in \mathbb{R}_+^2,
\end{aligned} \tag{4}$$

given,

$$g_0^{obj} = x_0 + 0.64x_1.$$

$$g_1^{obj} = 1.11x_0 + 0.76x_1.$$

$$g_2^{obj} = 1.11x_0 + 0.85x_1.$$

Its formulation in the CBF format is written in verbatim.

```

# File written using this version of the Conic Benchmark Format:
#       | Version 1.
VER
1

# The sense of the objective is:
#       | Maximize.
OBJSENSE
MAX

# Two scalar variables in this one conic domain:
#       | Two are nonnegative.
VAR
2 1
L+ 2

# Two scalar constraints with affine expressions in these two conic domains:
#       | One is in the nonpositive domain.
#       | One is in the nonnegative domain.
CON
2 2
L- 1
L+ 1

# Two coordinates in a^{obj}_j coefficients:
#       | a^{obj}[0] = 1.0

```

```

#      | a^{obj}[1] = 0.64
OBJCOORD
2
0 1.0
1 0.64

# Four coordinates in a_ij coefficients:
#      | a[0,0] = 50.0
#      | a[1,0] = 3.0
#      | and more...
ACCOORD
4
0 0 50.0
1 0 3.0
0 1 31.0
1 1 -2.0

# Two coordinates in b_i coefficients:
#      | b[0] = -250.0
#      | b[1] = 4.0
BCCOORD
2
0 -250.0
1 4.0

# New problem instance defined in terms of changes.
CHANGE

# Two coordinate changes in a^{obj}_j coefficients. Now it is:
#      | a^{obj}[0] = 1.11
#      | a^{obj}[1] = 0.76
OBJCOORD
2
0 1.11
1 0.76

# New problem instance defined in terms of changes.
CHANGE

# One coordinate change in a^{obj}_j coefficients. Now it is:
#      | a^{obj}[0] = 1.11
#      | a^{obj}[1] = 0.85
OBJCOORD
1
1 0.85

```